

APPLICATION
FOR
UNITED STATES LETTERS PATENT

TITLE: SERIAL MEDIA INDEPENDENT INTERFACE WITH
DOUBLE DATA RATE

APPLICANT: WILLIAM LO AND NAFEA BISHARA

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL584939235US

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the U.S. Patent and Trademark Office, P.O. Box 2327, Arlington, VA 22202.

December 5, 2001

Date of Deposit

Signature

Gabe Lewis

Typed or Printed Name of Person Signing Certificate

SERIAL MEDIA INDEPENDENT INTERFACE WITH DOUBLE DATA RATE**TECHNICAL FIELD**

This invention relates to network interfaces, and more particularly to serial media independent interfaces (SMII).

BACKGROUND

5 As computer systems continue to evolve, an increasing number of computers are interconnected in local area networks that are based on the Ethernet standard. Ethernet networks may employ different types of physical media such as twisted copper, fibre, 10 Mbit, and 100 Mbit to physically interconnect the
10 computers. The media independent interface (MII) is a specification that defines a standard interface for flow control and data transfer between a media access control layer (MAC) and any of the physical layers (PHY) that interface with the physical media of an Ethernet network. The MII has evolved to
15 include a reduced media independent interface (RMII) that reduced the pin-count of the interface to permit smaller, lower cost devices. The MII has further evolved beyond the RMII to include a serial-MII (SMII) specification that further reduces pin-count. SMII allows multi-port communication with a single
20 system clock. However, SMII requires two pins per port to convey complete MII information between a PHY and a MAC.

SMII 16 that may include a MAC component 16a and a PHY component 16b. The network interface circuit 10 interfaces one or more Ethernet network ports to a computer (not shown). The network interface circuit 10 may be implemented on a peripheral device such as a network interface card and as an integral portion of the computer such as on a motherboard of the computer. The double data rate SMII 16 supports Ethernet 10/100 physical layers and may communicate complete MII information between the MAC 12 and the PHY 14. The SMII 16 provides unidirectional communication between the MAC 12 and PHY 14 through one or more ports and advantageously only requires an average of one pin per port. In a conventional unidirectional system, two pins for port would be required, one pin for transmit and one pin for receive. Instead, the SMII interleaves transmit signals from pairs of ports through one pin, and interleaves receive signals from the pairs of ports on other pins, so that pairs of ports share two pins to communicate receive and transmit data. Therefore, by sharing pins between ports, an average of one pin per port is required to support multiple ports. Requiring only a single pin per port instead of the two pins per port required by conventional SMII significantly reduces the pin count required for the MAC 12 and PHY 14, permitting an increase in the quantity of Ethernet ports that are supported by each within given device profiles. For example, a MAC or PHY used for a 24

port hub would require 24 fewer pins without eliminating functionality.

The double data rate SMII 16 only requires a single clock 18 to maintain communication between the MAC 12 and the PHY 14. The clock 18 preferably operates at approximately 125 MHz. However, the clock frequency is not limiting and other frequencies both greater than and less than 125 MHz may be used. The double data rate SMII 16 is preferably included within the PHY 14 and MAC 12 so that the advantages of reduced pin count can be used to either reduce package size or increase the quantity of ports that are supported by the PHY 14 and MAC 12. Additional PHYs 20 or MACs may be operated from the same clock 18 to further increase the quantity of ports that are supported by the double data rate SMII 16.

Figure 2 illustrates interleaving a 10 bit segment 24 from an even port with a 10 bit segment 25 from an odd port to form a 20 bit segment 26 that is communicated through a single pin between a MAC and a device such as a PHY or another MAC. The 20 bit segment 26 is communicated through the single pin at twice the frequency of the 10 bit segments 24 and 25. The 20 bit segment 26 is then separated into two 10 bit segments 27 and 28.

Figure 3A shows a receive sequence diagram for the double data rate SMII 16. The receive sequence diagram depicts the relation between received bits RXD 30, a RX_CLK 32, and an

RX_SYNC 34. The received bits are latched in on both the positive-going clock edges and the negative-going clock edges. The received bits 30 are sent as 20 bit segments. The RX_SYNC 34 is generated by the PHY 14 every 10 clock cycles to delimit the boundaries of the bit segments.

Figure 3B shows a transmit sequence diagram for the double data rate SMII 16. The transmit sequence diagram depicts the relation between transmitted bits TXD 36, a REF_CLK 38, and a TX_SYNC 40. The transmitted bits are sampled on both the positive-going clock edges and the negative-going clock edges. The transmitted bits 36 are sent as 20 bit segments. The TX_SYNC 40 is generated by the MAC 12 every 10 clock cycles to delimit the boundaries of the bit segments. The PHY 14 preferably delimits the segments based on the positive-going edge of the TX_SYNC 40 and ignores the negative-going edge of TX_SYNC 40.

Figure 4A shows an embodiment of a PHY transmit circuit 50 portion of the double data rate SMII 16. The transmit circuit 50 uses a clock signal having a first operating frequency, such as 125 MHz, to generate two data streams that each have a frequency that is equal to the first operating frequency from a data stream having a frequency that is twice the first operating frequency. Inputting the data stream at about twice the first operating frequency permits a single pin to be used per port.

Data may be latched using both the rising-edge and the falling edge of the clock signal to generate the lower frequency data streams.

The PHY transmit circuit 50 receives the REF_CLK 38 and transmit data, TXD, 36 from the MAC 12. The REF_CLK 38 is input to a delay circuit 52 that generates a clock signal output that is delayed a quarter cycle. The output of the delay circuit 52 is coupled to latches 54-58 to generate TXD_EVEN 60 and TXD_ODD 62 from TXD 36. TXD_EVEN 60 and TXD_ODD 62 may be processed by standard physical layer techniques to generate the transmitted Ethernet compliant signal. An inverter 64 generates the SMII_REF_CLK 66 from the delay circuit output.

Figure 4B shows an embodiment of a PHY receive circuit 70 portion of the double data rate SMII 16. The PHY receive circuit 70 uses a clock signal having a first operating frequency, such as 125 MHz, to generate a data stream having a frequency that is twice the first operating frequency from two data streams that each have an operating frequency that is equal to the first operating frequency. Generating the data stream with a frequency that is twice the first operating frequency permits a single pin to be used per port. Data may be latched in using both the rising-edge and the falling edge of the clock signal to generate the higher frequency data stream.

162 from RXD 136. RXD_EVEN 160 and RXD_ODD 162 may be processed by standard MAC layer techniques. An inverter 164 generates the SMII_RX_CLK 166 from the RX_CLK 138.

Figure 5B shows an embodiment of a MAC transmit circuit 170 portion of the double data rate SMII 16. The MAC transmit circuit 170 uses a clock signal having a first operating frequency, such as 125 MHz, to generate a data stream having a frequency that is twice the first operating frequency from two data streams that each have an operating frequency that is equal to the first operating frequency. Generating the data stream with a frequency that is twice the first operating frequency permits a single pin to be shared by two ports. Data may be latched in using both the rising-edge and the falling edge of the clock signal to generate the higher frequency data stream.

The MAC transmit circuit 170 receives the SMII_REF_CLK 172 and two receive data streams, TXD_EVEN 174 and TXD_ODD 176, from processing circuits within the MAC 12. The SMII_REF_CLK 172 is input to a delay circuit 178 and latches 181-183. A buffer 180 coupled to the output of the delay circuit 178 generates REF_CLK 132. Latches 181-183 latch in data from TXD_EVEN 174 and TXD_ODD 176. The delay circuit 178 generates a clock signal output that is delayed a quarter cycle. A combiner 184 combines latched data from TXD_EVEN 174 and TXD_ODD 176 to generate TXD

130. The MAC transmit circuit 170 transmits the TX_CLK 138 and transmit data, TXD, 130 to the PHY 14.

Figure 6 shows timing diagrams for outputs and inputs of the PHY 14. The PHY inputs show the timing relation between the REF_CLK 90 and the TXD and TX_SYNC 92. For the PHY inputs the preferable values for Tsetup and Thold are -0.9 nsec and 2.7 nsec respectively. The PHY outputs show the timing relation between the RX_CLK 94 and RXD and RX_SYNC 96. For the PHY outputs the preferable values for Tsetup and Thold are 1.4 nsec and 1.2 nsec respectively. The duty cycle of RX_CLK is preferably 3.6 nsec minimum and 4.4 nsec maximum.

Figure 7 shows timing diagrams for outputs and inputs of the MAC 12. The MAC outputs show the timing relation between the REF_CLK 100 and the TXD and TX_SYNC inputs 102. For the MAC outputs the preferable value for Tskew is 0.5 nsec. The MAC inputs show the timing relation between the RX_CLK 104 and the RXD and RX_SYNC 106. For the MAC inputs the preferable values for Tsetup and Thold are 1.0 nsec and 0.8 nsec respectively. The duty cycle of RX_CLK is preferably 3.6 nsec minimum and 4.4 nsec maximum.

A number of embodiments of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, the interface may

